

**TITLE**

**METHOD FOR UPDATING COEFFICIENTS IN DECISION FEEDBACK**  
**EQUALIZER**

**BACKGROUND OF THE INVENTION**

**5 Field of the Invention:**

The present invention relates to a decision feedback equalizer and particularly to a method for generating a Toeplitz matrix in a decision feedback equalizer to reduce inter-chip interference (ICI) and inter-symbol interference  
10 (ISI).

**Description of the Related Art:**

Wireless communication channels often suffer from severe inter-chip interference (ICI), inter-symbol interference (ISI), and multi-path fading effect. Channel  
15 equalization is essential to combat channel distortion caused by the above, so that symbols can be correctly determined at the receiving end. Channel equalization can be performed either on a chip basis or on a sample basis. In chip based equalization, the equalizer coefficients are  
20 updated on every received chip. Decision feedback equalizer (DFE) using the least mean-squared (LMS) algorithm is a well-known equalization technique for magnetic storage and digital communication. The basic block diagram of traditional DFE is depicted in Fig. 1, where DFE is composed  
25 of two main finite impulse response (FIR) filters, the feedforward filter (FFF) 102 and the feedback filter (FBF) 104. The outputs from both filters are added and fed into a slicer 106. The signal output from the slicer 106 is the

final equalized data. The basic function of FFF 102 and FBF 104 is to cancel the pre-cursor and post-cursor inter-chip interference (ICI) respectively, while the WUC 108 and WUD 110 in the figure stand for the weight-update blocks for the 5 FFF 102 and the FBF 104.

DFE in the chip-base helps alleviate ICI. ISI, however, cannot be completely cancelled with this type of DFE. Thus, a DFE that reduces ISI and ICI simultaneously is desirable.

10

#### **SUMMARY OF THE INVENTION**

The object of the present invention is to provide a method for generating a Toeplitz matrix in a decision feedback equalizer (DFE). A DFE of this type is then capable of reducing inter-symbol interference (ISI) and 15 inter-chip interference (ICI) simultaneously.

The present invention provides a method for updating coefficients in a decision feedback equalizer (DFE). The DFE comprises an ISI canceller for canceling a ISI from a plurality of first signals received from a channel. A first 20 symbol comprising a set of the first signals is decoded to generate a decoded symbol. Then, a vector of error values computed as the difference between the decoded symbol, and the first symbol is obtained. According to the decoded symbol and the vector of the error values, a temp matrix is 25 generated. Next, the values of the elements in every diagonal line of the temp matrix are averaged. Finally, the coefficients are updated by the Toeplitz Matrix.

The present invention also provides a decision feedback equalizer (DFE). The DFE comprises an ICI canceller and an

ISI canceller. The ICI canceller cancels ICI from a signal received from a channel and outputs a first signal without ICI. The ISI canceller comprises a symbol decoder and a symbol-base feedback filter with a plurality of 5 coefficients. A first symbol comprising a set of the first signals is decoded by the symbol decoder. The symbol-base feedback filter transforms the decoded symbol by a Toeplitz Matrix to cancel ISI from the present decoded symbol, and generates an output signal. The Toeplitz Matrix is 10 generated by the above method.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

The present invention will become more fully understood from the detailed description given hereinbelow and the accompanying drawings, given by way of illustration only and 15 thus not intended to be limitative of the present invention.

Fig. 1 is a block diagram illustrating a configuration of a traditional DFE.

Fig. 2 is a block diagram illustrating a configuration of a DFE according to the first embodiment of the present 20 invention.

Fig. 3 is a flow chart illustrating the method for updating coefficients in the DFE according to the present invention.

#### **DETAILED DESCRIPTION OF THE INVENTION**

25 Fig. 2 is a block diagram illustrating a configuration of a DFE according to the first embodiment of the present invention. The DFE comprises an inter-chip interference (ICI) canceller 200 and an inter-symbol interference (ISI)

canceller 250. A signal  $X_k(n)$  received from a channel is input to the ICI canceller 200.

The ICI canceller 200 is composed of two main chip-base finite impulse response (FIR) filters, the feedforward filter (FFF) 210 and the feedback filter (FBF) 220. The basic function of FFF 210 and FBF 220 is to cancel the precursor and post-cursor inter-chip interference (ICI) respectively. The FFF 210 and FBF 220 adapt the tap weights to compensate for ICI and other sources of signal degradation and distortion.

The outputs from both filters  $Y_{cf}$  and  $Y_{cd}$  are added by an adder 240 to become input data  $Y_c$ . The input data  $Y_c$  is fed into a hard decision block 230. Then, the hard decision block 230 produces equalized data  $HD(Y_c)$ . After a one-tap delay block 246, the equalized data  $HD(Y_c)$  is fed into FEF 220.

On the other hand, an error  $E_c$  defines a vector of error values computed as the difference between an output  $C_k$  of a multiplexer 244 and the input data  $Y_c$ . In the embodiment, the DFE can operate in a Barker mode or a CCK mode. The operating mode can be selected by control signals  $S_{c1}$  and  $S_{c2}$  input to the multiplexer 244.

Then, the error  $E_c$  is fed back to the FFF 210 and FBF 220 to adapt the tap weights. The WUC 212 and WUD 222 in the figure stand for the weight-update blocks for the FFF 210 and the FBF 220.

In this embodiment of the invention, the WUC 212 and WUD 222 use Least-Mean-Square(LMS) to adapt the tap weights as is shown in the following example.

For the Barker mode, the adaptive algorithm is:

$$W_k(n+1) = W_k(n) + i * \text{conj}(E_c(n)) * X(n-k), \quad k=0, 1, \dots, N; \quad (1)$$

where  $W_k(n+1)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FFF 210 after a period of time, i.e. the vector set formed by the adjusted tap weights;

5  $W(n)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FFF 210 at a chip time  $n$ ; and  
i is the learning constant.

$$B_k(n+1) = B_k(n) + i * \text{conj}(E_c(n)) * B_k[Y_c(n-k)], \quad k=0, 1, \dots, M; \quad (2)$$

10 where  $B_k(n+1)$  is the vector set formed by the tap weights  $B_1, B_2, \dots, B_M$  in the FBF 220 after a period of time, i.e. the vector set formed by the adjusted tap weights;

$B_k(n)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FBF 220 at a chip time  $n$ ; and  
i is the learning constant.

15 The input data  $Y_c(n)$  fed into a hard decision block 230 is :

$$Y_c(n) = \sum_{k=0}^N X_k(n-k) \text{conj}(W_k) - \sum_{k=0}^M H_k[Y_c(n-k-1)] \text{conj}(B_k) \quad (3)$$

The error  $E_c(n)$  is :

$$E_c(n) = C_k(n) - Y_c(n) \quad (4)$$

20 For the CCK mode, the adaptive algorithm is:

$$W_k(n+1) = W_k(n) + i * \text{conj}(E_c(n-16)) * X(n-k-16), \quad k=0, 1, \dots, N; \quad (5)$$

where  $W_k(n+1)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FFF 210 after a period of time, i.e. the vector set formed by the adjusted tap weights;

25  $W(n)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FFF 210; and  
i is the learning constant.

$$B_k(n+1) = B_k(n) + i * \text{conj}(E_c(n-16)) * B_k[Y_c(n-k-17)], \\ k=0, 1, \dots, M; \quad (6)$$

where  $B_k(n+1)$  is the vector set formed by the tap weights  $B_1, W_2, \dots, B_M$  in the FBF 220 after a period of time, i.e. the vector set formed by the adjusted tap weights;

$B_k(n)$  is the vector set formed by the tap weights  $W_1, W_2, \dots, W_N$  in the FBF 220; and  
5  $i$  is the learning constant.

The input data  $Y_c(n)$  fed into a hard decision block 230 is :

$$Y_c(n) = \sum_{k=0}^N X_k(n-k) \text{conj}(W_k) - \sum_{k=0}^M \text{HD}[Y_c(n-k-1)]_k \text{conj}(B_k) \quad (7)$$

10 The error  $E_c(n)$  is :

$$E_c(n-16) = C_k(n-16) - Y_c(n-16) \quad (8)$$

After a symbol time delay block, which is 8-tap delay block 260 in the embodiment, a symbol  $Y_{cs}$  which comprises eight input data  $Y_c$  is fed in to an adder 270. The ISI 15 canceller 250 is composed from a symbol-based feedback filter (FBF) 270 and a CCK or barker decoder 280. The symbol-based FBF 270 adapts the tap weights to compensate for ISI generated by the last symbol. The outputs from the symbol-based FBF 270  $Y_{bs}$  and the symbol  $Y_{cs}$  are added by an 20 adder 290 to become an input symbol  $Y_s$ .

The input symbol  $Y_s$  is fed into the CCK or barker decoder 280. Then, the CCK or barker decoder 280 produces a decoded symbol  $C(Y_s)$ . An error  $E_s$  defines a vector of error values computed as the difference between the decoded symbol 25  $C(Y_s)$  and the input symbol  $Y_s$  after a one-tap delay block 294.

Then, the error  $E_c$  and the input symbol  $Y_s$  after a one-tap delay block 292 is input to a Toeplitz matrix block 272.

The TMB 272 in the figure stands for the coefficient-update block for the symbol-based FBF 270.

In this embodiment of the invention, the TMB 272 uses Least-Mean-Square(LMS) to adapt the tap weights, for  
5 example.

The adaptive algorithm is:

$$H(m+1) = H(m) + \mu T \{ \text{conj}(E(m)) \bullet C(m+1) \}; \quad (9)$$

where  $H(m)$  is coefficients at a symbol time  $m$ ;

$H(m+1)$  is coefficients at a symbol time  $(m+1)$ ;

10  $\mu$  is a predetermined gain;

$T$  is a Toeplitz Matrix;

$E(m)$  is the vector of error values; and

$C(m+1)$  is the decoded symbol at the symbol time  $(m+1)$ .

The symbol  $Y_{cs}(m)$  which comprises eight input data  $Y_c(n-15)$ ,  $Y_c(n-14)$ ,  $Y_c(n-13)$ ,  $Y_c(n-12)$ ,  $Y_c(n-11)$ ,  
15  $Y_c(n-10)$ ,  $Y_c(n-9)$ ,  $Y_c(n-8)$  is take as an example, where  $m$  is a symbol time and a natural number, and  $n=8 \times m$ . The algorithm to compensate for ISI generated by the last symbol  $C(Y_s(m-1))$  is :

$$\begin{aligned} 20 Y_s(m) &= Y_{cs}(m) - C(Y_s(m-1)) \bullet H(m) \\ &= Y_{cs}(m) - [C(n-15) \ C(n-14) \ C(n-13) \ C(n-12) \ C(n-11) \\ &\quad C(n-10) \ C(n-9) \ C(n-8)] \bullet H(m) \end{aligned} \quad (10)$$

The following illustrates how the Toeplitz Matrix in the TMB 272 is generated.

First, the symbol  $Y_{cs}(m)$  which comprises input data  
25  $Y_c(n-7)$ ,  $Y_c(n-6)$ ,  $Y_c(n-5)$ ,  $Y_c(n-4)$ ,  $Y_c(n-3)$ ,  $Y_c(n-3)$ ,  $Y_c(n-1)$ ,  $Y_c(n)$  is decoded to generate a decoded symbol  $C(Y_{cs}(m))$ .

An error  $E_s(m)$  defines a vector of error values computed as the difference between the decoded symbol  $C(Y_{cs}(m))$  and the input symbol  $Y_{cs}(m)$  after a one-tap delay  
30 block 294. Then, according to the decoded symbol

Client's ref. : VITO3 0097  
 Our ref: 0608-9902 usf/ellenstove

$C(Y_{cs}(m))$  and the vector of the error values  $E_s(m)$ , a temp

Matrix  $T(m)$  
$$\begin{bmatrix} h_{11} & h_{12} & \cdots & \cdots & h_{18} \\ h_{21} & h_{22} & \cdots & \cdots & h_{28} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_{71} & h_{72} & \cdots & \cdots & h_{78} \\ h_{81} & h_{82} & \cdots & \cdots & h_{88} \end{bmatrix}$$
 is generated. The temp

Matrix  $T(m)$  is :

$$T(m) = \begin{bmatrix} E^*(n-7) \cdot C(n-7) & E^*(n-7) \cdot C(n-6) & \cdots & \cdots & E^*(n-7) \cdot C(n) \\ E^*(n-6) \cdot C(n-7) & E^*(n-6) \cdot C(n-6) & \cdots & \cdots & E^*(n-6) \cdot C(n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ E^*(n-1) \cdot C(n-7) & E^*(n-1) \cdot C(n-6) & \cdots & \cdots & E^*(n-1) \cdot C(n) \\ E^*(n) \cdot C(n-7) & E^*(n) \cdot C(n-6) & \cdots & \cdots & E^*(n) \cdot C(n) \end{bmatrix};$$

5 where  $E(n)$  is a vector of error values at the chip time  $n$  and  $C(n)$  is the chip of the decoded symbol at the chip time  $n$ .

If the channel is steady, such as the DFE design for wireless communication over mobile radio channel, the values 10 of the elements in the diagonal lines of the temp Matrix are almost the same i.e.  $h_{11}=h_{22}=\dots=h_{88}$ ,  $h_{21}=h_{32}=\dots=h_{87}, \dots, h_{71}=h_{82}$ ,  $h_{12}=h_{23}=\dots=h_{78}$ ,  $h_{13}=h_{24}=\dots=h_{77}, \dots, h_{17}=h_{28}$ .

Thus, for reducing noise, the values of the elements in every diagonal line of the temp matrix are averaged to

15 generate a Toeplitz Matrix  $H(m)$  
$$\begin{bmatrix} h_8 & h_9 & \cdots & \cdots & h_{15} \\ h_7 & h_8 & \cdots & \cdots & h_{14} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2 & h_3 & \cdots & \cdots & h_9 \\ h_1 & h_2 & \cdots & \cdots & h_8 \end{bmatrix}.$$
 The

Toeplitz Matrix  $H(m) =$

$$\begin{bmatrix} (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 & (\sum_{i=0}^6 E(n-(i+1)) \cdot Q(n-i))/7 & \dots & \dots & E(n-7) \cdot Q(n) \\ (\sum_{i=0}^6 E(n-i) \cdot Q(n-(i+1))/7 & (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 & \dots & \dots & (\sum_{i=0}^1 E(n-(i+6)) \cdot Q(n-i))/2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\sum_{i=0}^1 E(n-i) \cdot Q(n-(i+6))/2 & (\sum_{i=0}^2 E(n-i) \cdot Q(n-(i+5))/3 & \dots & \dots & (\sum_{i=0}^6 E(n-(i+1)) \cdot Q(n-i))/7 \\ E(n) \cdot Q(n-7) & (\sum_{i=0}^1 E(n-i) \cdot Q(n-(i+6))/2 & \dots & \dots & (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 \end{bmatrix}.$$

In addition, in the Toeplitz Matrix

$$\begin{bmatrix} h_8 & h_9 & \dots & \dots & h_{15} \\ h_7 & h_8 & \dots & \dots & h_{14} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2 & h_3 & \dots & \dots & h_9 \\ h_1 & h_2 & \dots & \dots & h_8 \end{bmatrix}, \text{ for any } 15 \geq i > 8, \text{ the } h_i \dots h_{15} \text{ may be equal}$$

to 0, such as, if  $h_9 \dots h_{15}$  are small and can be ignored, the

5 Toeplitz Matrix becomes  $\begin{bmatrix} h_8 & 0 & \dots & \dots & 0 \\ h_7 & h_8 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ h_2 & h_3 & \dots & \dots & 0 \\ h_1 & h_2 & \dots & \dots & h_8 \end{bmatrix}$ . Thus, the

Toeplitz Matrix  $H(m) =$

$$\begin{bmatrix} (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 & 0 & \dots & \dots & 0 \\ (\sum_{i=0}^6 E(n-i) \cdot Q(n-(i+1))/7 & (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ (\sum_{i=0}^1 E(n-i) \cdot Q(n-(i+6))/2 & (\sum_{i=0}^2 E(n-i) \cdot Q(n-(i+5))/3 & \dots & \dots & 0 \\ E(n) \cdot Q(n-7) & (\sum_{i=0}^1 E(n-i) \cdot Q(n-(i+6))/2 & \dots & \dots & (\sum_{i=0}^7 E(n-i) \cdot Q(n-i))/8 \end{bmatrix}.$$

In summary, the flow chart illustrating the method for updating coefficients in the DFE according to the present  
 10 invention is shown in Fig. 3. First, a first symbol

comprising a set of the first signals is decoded to generate a decoded symbol (S302). Then, a vector of error values computed as the difference between the decoded symbol, and the first symbol is obtained (S304). According to the 5 decoded symbol and the vector of the error values, a temp matrix is generated (S306). Next, the values of the elements in every diagonal line of the temp matrix are averaged (S308). Finally, the coefficients are updated by the Toeplitz Matrix (S310).

10 The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. Obvious modifications or variations are possible in light of the above teaching. The embodiments were chosen and described to provide the best 15 illustration of the principles of this invention and its practical application to thereby enable those skilled in the art to utilize the invention in various embodiments and with various modifications as are suited to the particular use contemplated. All such modifications and variations are 20 within the scope of the present invention as determined by the appended claims when interpreted in accordance with the breadth to which they are fairly, legally, and equitably entitled.